Figure 1

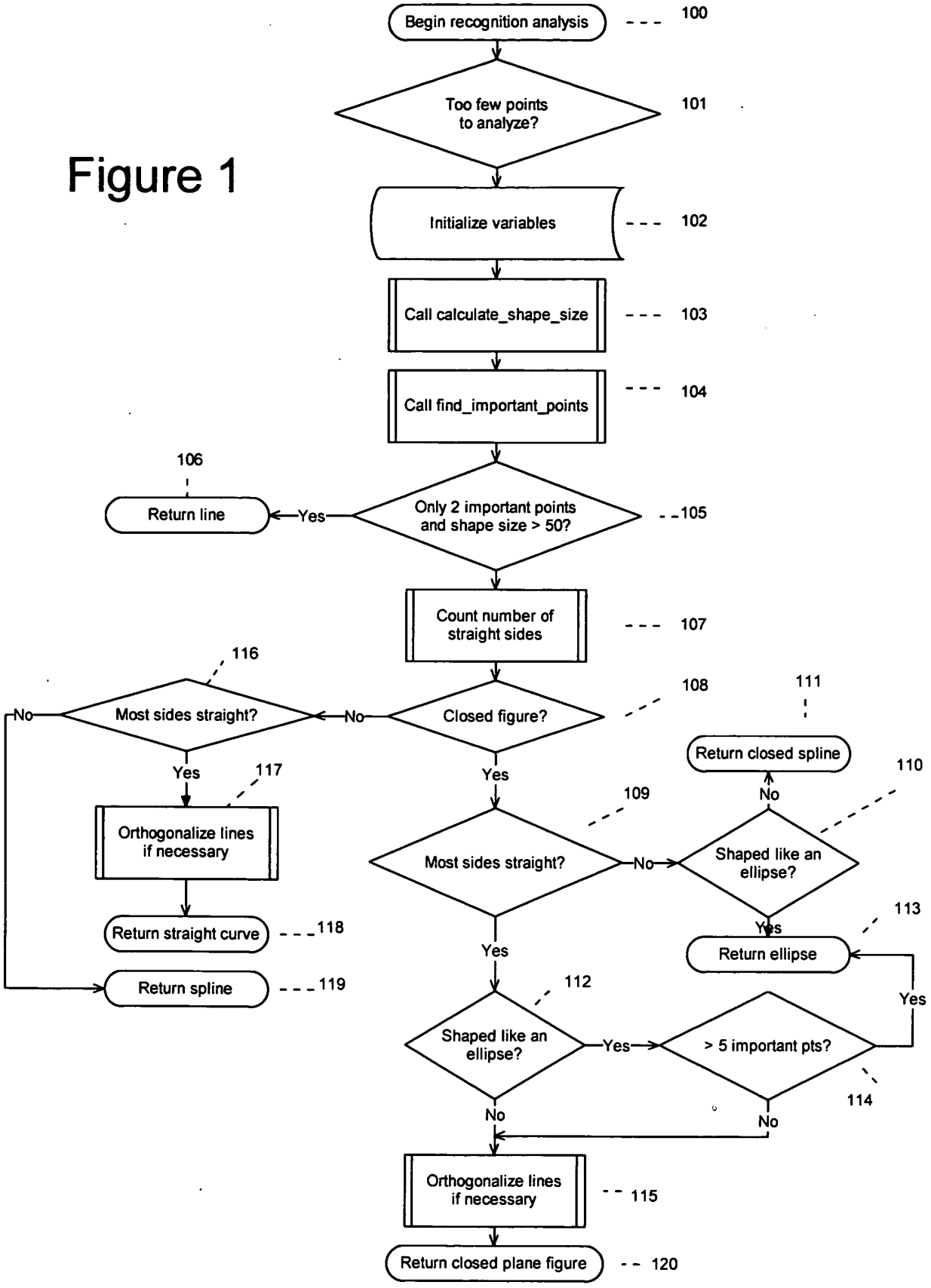

Begin recognition analysis
100
Too few points to analyze?
101
Initialize variables
102
Call calculate_shape_size
103
Call find_important_points
104
106
Return line
Yes
Only 2 important points and shape size > 50?
105
Count number of straight sides
107
116
No
Most sides straight?
No
Closed figure?
108
111
Return closed spline
Yes
117
Yes
109
No
110
Orthogonalize lines if necessary
Most sides straight?
No
Shaped like an ellipse?
Return straight curve
118
Yes
113
Return ellipse
Return spline
119
Yes
112
Yes
Shaped like an ellipse?
Yes
> 5 important pts?
Yes
114
No
No
Orthogonalize lines if necessary
115
Return closed plane figure
120

Figure 2

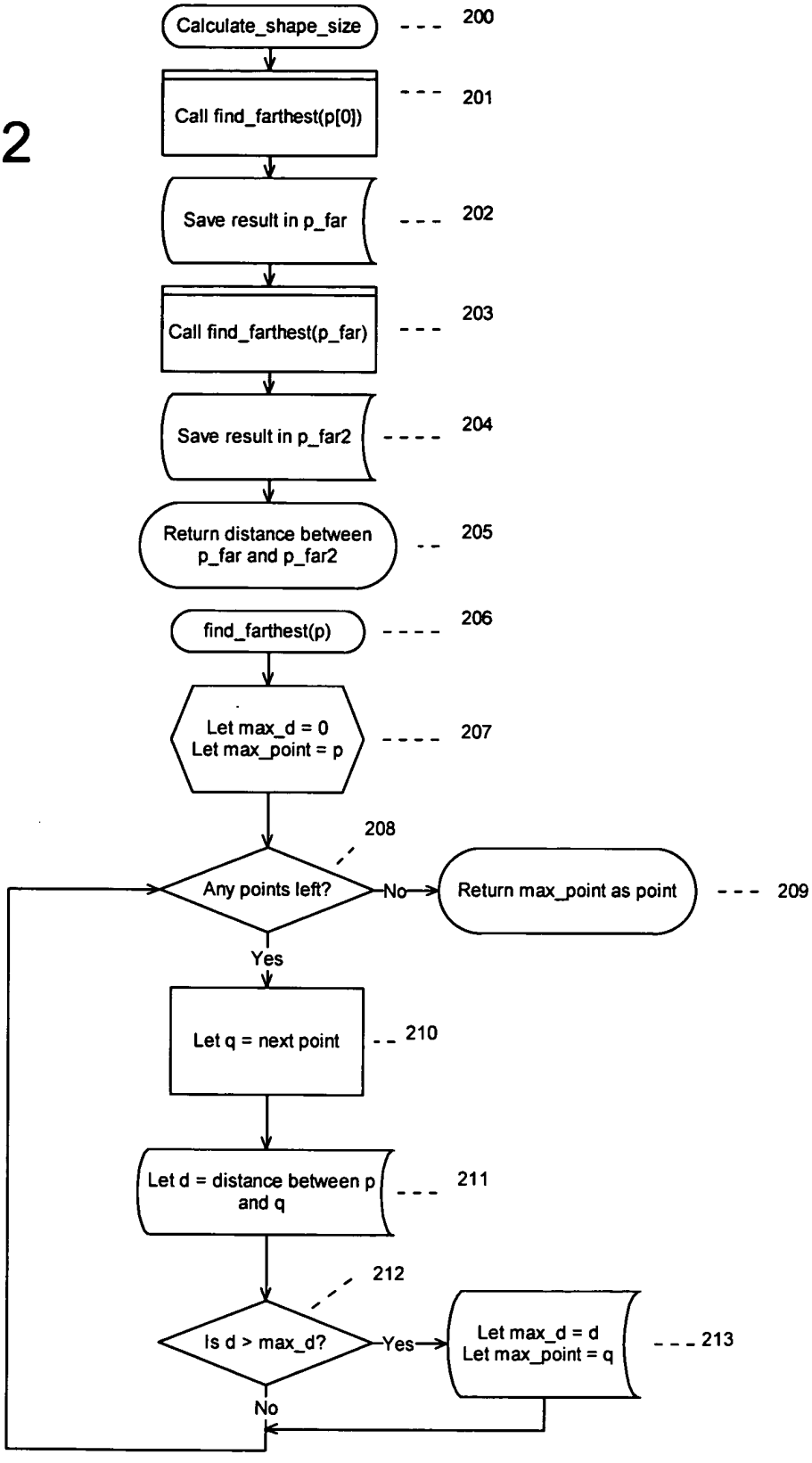
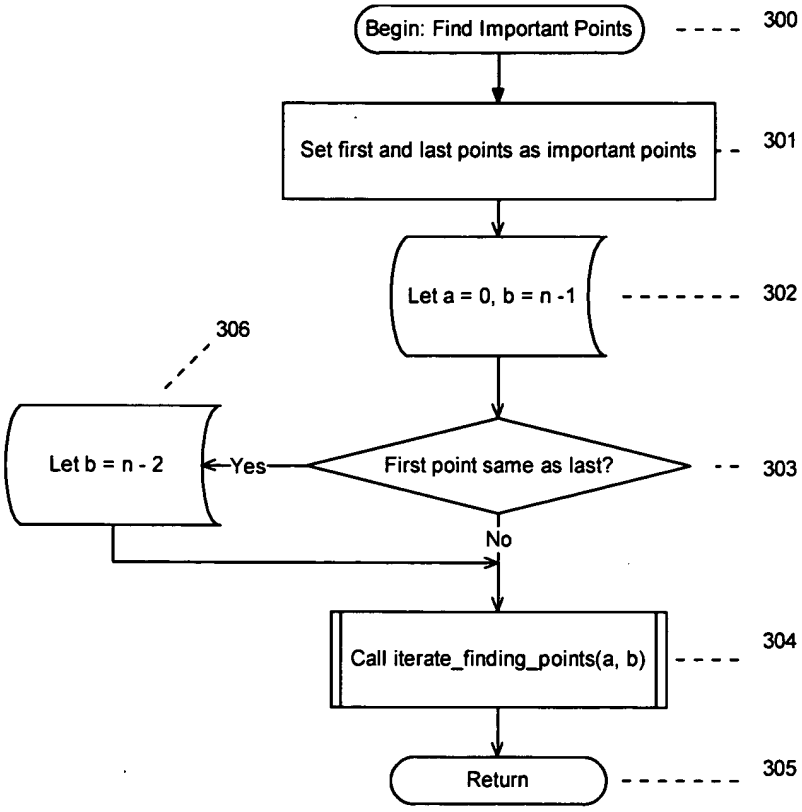

Calculate_shape_size
200
Call find_farthest(p[0])
201
Save result in p_far
202
Call find_farthest(p_far)
203
Save result in p_far2
204
Return distance between p_far and p_far2
205
find_farthest(p)
206
Let max_d = 0
Let max_point = p
207
208
Any points left?
No
Return max_point as point
209
Yes
Let q = next point
210
Let d = distance between p and q
211
212
Is d > max_d?
Yes
Let max_d = d
Let max_point = q
213
No

Figure 3



Copyright Joshua Goldfoot

Figure 4

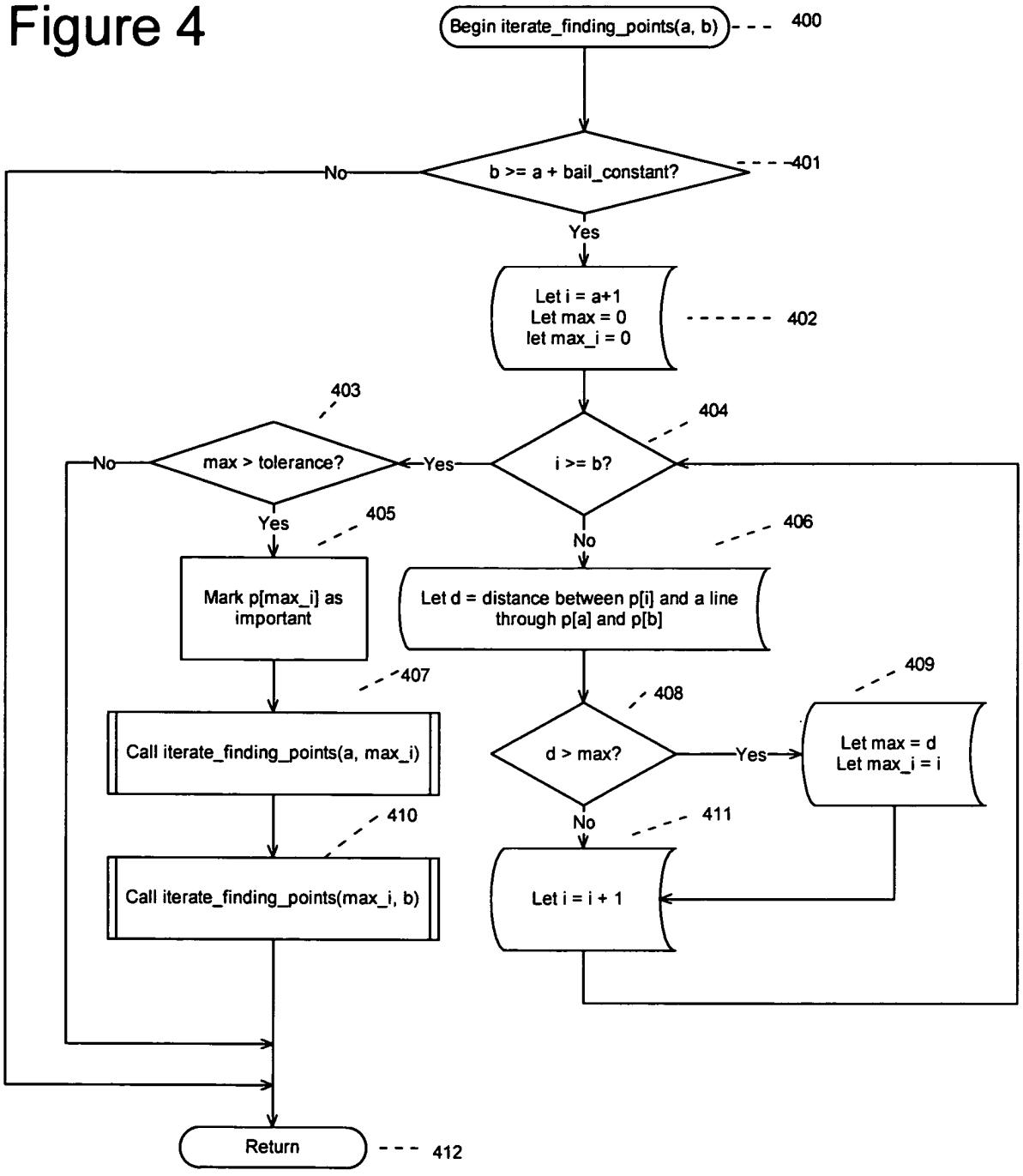
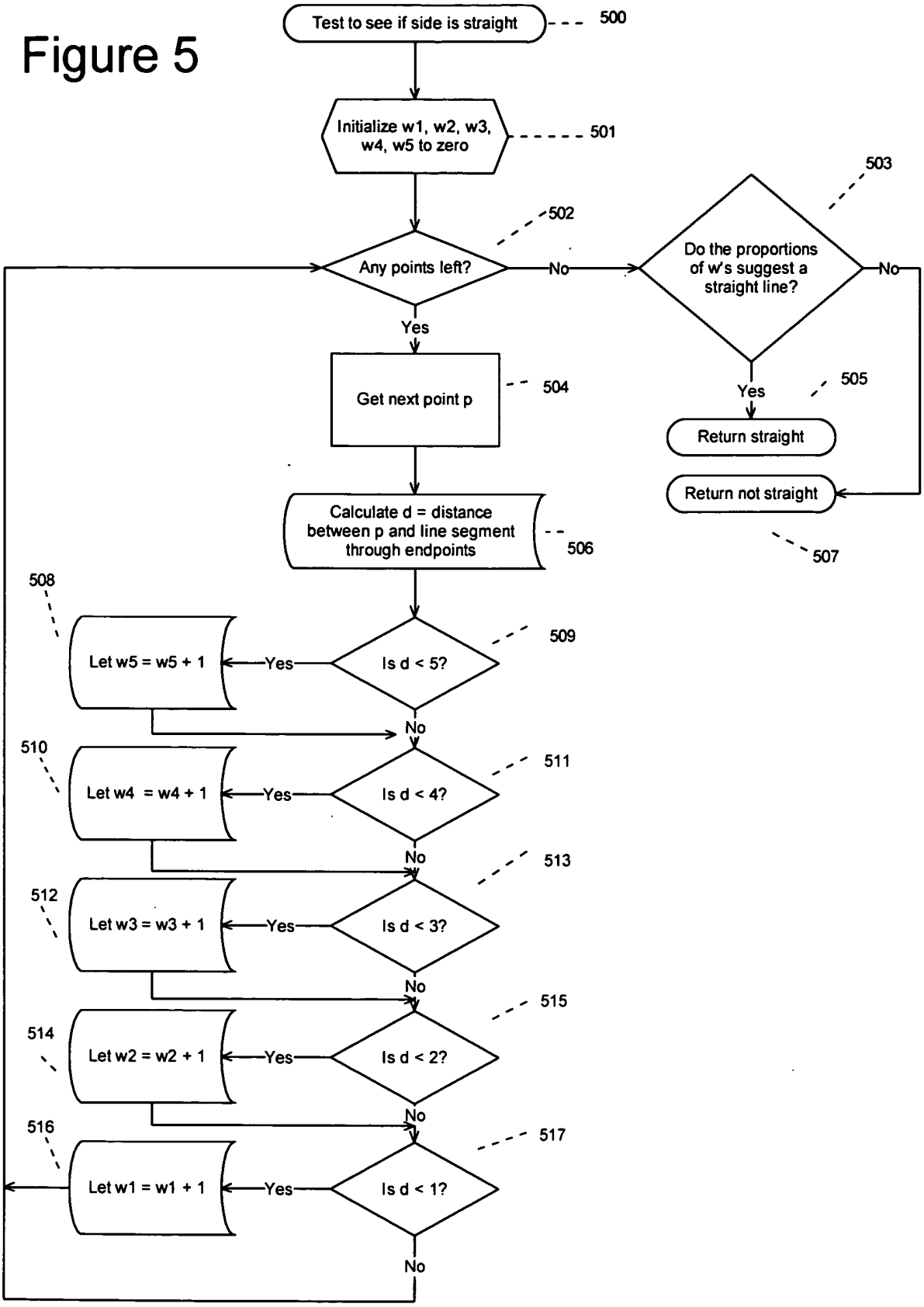
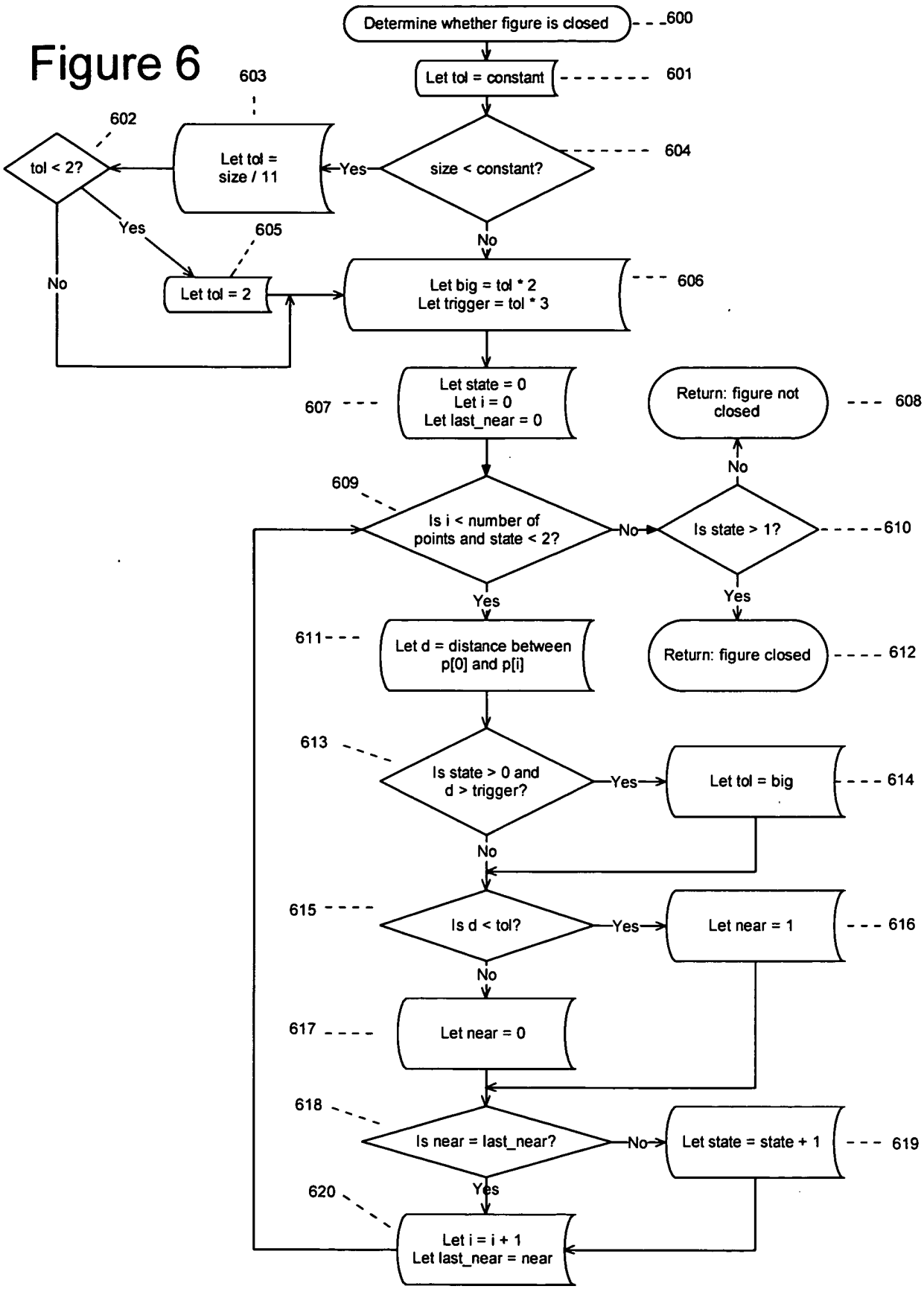

Begin iterate_finding_points(a, b)
400
b >= a + bail_constant?
401
No
Yes
Let i = a+1
Let max = 0
let max_i = 0
402
403
404
max > tolerance?
i >= b?
No
Yes
Yes
No
405
406
Mark p[max_i] as important
Let d = distance between p[i] and a line through p[a] and p[b]
407
408
409
Call iterate_finding_points(a, max_i)
d > max?
Yes
Let max = d
Let max_i = i
410
No
411
Call iterate_finding_points(max_i, b)
Let i = i + 1
Return
412

Figure 5



Document 1 - Page 5

Figure 6


Determine whether figure is closed
600
Let tol = constant
601
size < constant?
604
Yes
Let tol = size / 11
603
tol < 2?
602
Yes
No
Let tol = 2
605
No
Let big = tol * 2
Let trigger = tol * 3
606
Let state = 0
Let i = 0
Let last_near = 0
607
Return: figure not closed
608
No
Is i < number of points and state < 2?
609
No
Is state > 1?
610
Yes
Yes
Let d = distance between p[0] and p[i]
611
Return: figure closed
612
Is state > 0 and d > trigger?
613
Yes
Let tol = big
614
No
Is d < tol?
615
Yes
Let near = 1
616
No
Let near = 0
617
Is near = last_near?
618
No
Let state = state + 1
619
Yes
Let i = i + 1
Let last_near = near
620